METHOD OF AND APPARATUS FOR ASCERTAINING THE STATUS OF A DATA PROCESSING ENVIRONMENT

Technical Field

The present invention relates to a method of and apparatus for determining status of a data processing environment. The information concerning the status of the environment may include an indication of what devices are operating within the environment, what facilities they offer and whether the devices can be trusted.

Background Art

The issues of security and ease of use of a computing platform are often in conflict. For commercial applications, a client computing platform typically operates in an environment where its behaviour is vulnerable to modification. Such modification can be made by local or remote entities. This has given rise to concerns, especially in the field of e-commerce, that transactions conducted on a computer might be subject to some form of misdemeanour, such as theft of credit card details. These perceived insecurities may limit the willingness of users to undertake e-commerce transactions on either local or remote computer systems.

The data processing environment (or environment) of a computer platform or other data processing appliance consists of the other computing entities (computer platforms or any other data processing appliance) that are discrete from the computer platform and are in communication with it through one or more data networks. For a computer entity to form part of the environment of a computer platform, the computer platform must be able to interact with the entity but must not be constrained to do so – at some level, interaction must be voluntary. The boundary of an environment will generally be considered in terms of network connections (or other network "distance") between one point and another – for some purposes, the data processing environment of a computer platform may be considered its local network and no more, whereas for other purposes, the data processing environment of a computer platform may extend much further (for example, across a company intranet).

There are existing security applications, such as virus checkers and fire walls which can be installed in computer systems in order to limit their vulnerability to viruses or to malicious users seeking to take control of the machine remotely. However, these security applications execute on computing platforms under the assumption that the platform is operating as intended and that the platform itself will not subvert the processes used by these applications.

Users engaging in communication with a remote or unfamiliar data processing environment may nevertheless be concerned about the security of that environment as a whole rather than just the security of the computing device with which they have made initial contact. Thus users seek reassurance that the computing environment can be trusted.

As used herein, the word "trust" is used to mean that something can be trusted, in the sense that it is working in the way that it is intended and expected to work and is not or has not been tampered with in order to run malicious operations.

Disclosure of the Invention

According to a first aspect of the present invention, there is provided an apparatus for ascertaining the status of a data processing environment, comprising at least one trusted computing device which is arranged to challenge other devices within a data processing environment, to keep a record of the responses and to make the record available.

It is thus possible to use a trusted computing device to keep an audit of the status of a data processing network. The trusted device can challenge new devices as and when it discovers them within the data processing environment and can also re-challenge known devices every now and again in order to ascertain that they are functioning correctly. In such a challenge, a trusted computing device extracts from the challenged device a response to the challenge. Preferably the challenged device enters into a predefined challenge - response protocol, such as a TCPA challenge - response protocol in order to return trusted integrity and identity information about the challenged device. Thus the response may include information which can be analysed to determine the integrity of the challenged device, such as at least one integrity metric. The trusted device, upon receiving the response, extracts the integrity information from the response and compares it with an authenticated metric for the challenged device. As a result of the comparison, the trusted

device can indicate whether the challenged device can be trusted or not. By keeping a record of the time that a device is challenged, the response received from the device and the result of the comparison of the integrity metrics, the trusted computing device can maintain a log of the status of the data processing environment.

The integrity information would normally include a cryptographic representation of at least one platform component. This may, for example, include the BIOS, operating system or an application. Preferably the integrity information is of the form described in the TCPA specification (www.trustedpc.org) and has been measured and stored in the form described in the TCPA specification.

Advantageously other devices within the data processing environment can also issue challenges. The responses to those challenges and conclusions concerning trustworthiness can be recorded by the at least one trusted computing device. An indication of which devices acted as challenger and challengee can also be recorded, together with an indication of whether the challenger is itself established as trustworthy.

Preferably the challenges to known devices are made on a periodic basis in order to maintain an up to date record of the integrity of the data processing environment. However, additional challenges may also be made when a device attempts to perform an operation within the environment which requires that device to be trusted. Thus attempts to read, create or modify sensitive data, whether this data be user data, application data or system data (as defined by a system administrator) may require re-authentication of the trustworthiness of the device before it is enabled to have such access.

Advantageously the record held by the trusted device includes historical data representing the status of the network. Such data may be of use during investigations of system performance in order to indicate when a data processing environment could be regarded as trustworthy and/or what devices or processes may have entered or been executed in that environment. This information may be of use to administrators or investigators when seeking data concerning fraudulent transactions or attempts to subvert the operation of one or more components within the system.

In order to maintain a record of the devices within the computing environment, the trusted computing device needs to ascertain what devices are there. It can do this by sending query messages into the environment. The queries may be specific, that is directed to a known device in order to ascertain that it is still there and functioning. However, the queries may also be more general. Thus, for example, the trusted computing device could issue a query to a class of devices, for example printers, to identify what printers are available within the data processing environment. Such a query could then be repeated for a different class of device in order to build up a picture of the data processing environment. Alternatively, the trusted computing device could issue a general query asking devices to respond by identifying themselves, and optionally their functionality and/or integrity metrics. Advantageously the query also includes a generation identifier such that the age of the query can be ascertained by devices receiving the query. In this context, age can be measured in either or both the temporal sense or the number of retransmissions that the message has undergone. It is particularly desirable to limit the number of retransmissions that the query message may undergo as in extreme cases the message could propagate out of a local computing environment via, for example, a communications link to the internet and then computing devices receiving that query message could then attempt to respond. If this were the case, the trusted computing device could be overwhelmed by responses from computers which do not really constitute part of the data processing environment but which nevertheless have managed to receive the query message.

The trusted computing device can also listen to data communications on a network or between devices local to it in order to ascertain what devices are operating. The need to listen for devices entering and leaving the data processing network is likely to become more prevalent with the general increase in the number of portable computing devices and the ease at which these can enter or leave data processing environments as a result of the increase in wireless communication links to join computing devices, for example Blue Tooth.

When a user with a portable computing device or a remote user using a telecommunications link wishes to interact with a data processing environment, the user may seek to challenge the integrity of that environment. The functionality of the user's computing device and/or the communications bandwidth between the user's device and the

data processing network may limit the ability of the user to make individual challenges to each device in the data processing environment. However, the user may still seek confirmation that the data processing environment is secure, or at least an indication of the trust which he should place in that data processing environment (for a user may still decide to use a data processing environment even if that data processing environment is not deemed to be trustworthy - this is the user's choice depending on the type of transaction that the user wishes to undertake and the urgency ascribed to that transaction). With the present invention, a user does not need to make individual challenges to each device, but instead can seek this data from the trusted computing device. The user can trust the data from the trusted computing device because the user can challenge the trusted computing device and analyse the response to the challenge, comparing the integrity metrics received from the trusted computing device with those which are certificated as being the correct metrics, thereby enabling the user to determine whether the trusted computing device is itself trustworthy. The user can also determine what facilities are available in the computing environment.

According to a second aspect of the present invention, there is provided a computing device including a communications device and a data processor wherein the data processor is arranged to establish communication with a trusted computing device via the communication device, to receive at least part of the record of responses and to establish from an internal rules base whether the data processing environment is trustworthy enough to enable a class of transaction or task to be carried out in that environment.

According to a third aspect of the present invention, there is provided a computing device including a communications device and a data processor, wherein the computing device uses the communication device to establish communication with at least one device within a data processing system, and in which the data processor is arranged to identify challenges from at least one trusted computing device, to apply response rules to the challenge and, if a response is indicated, to respond to the challenge in accordance with the rules.

Advantageously, when the computing device receives a challenge from the trusted device it examines a generation identifier in order to see whether the message is still valid. The generation identifier may be a skip number. Thus, each time the challenge is retransmitted the skip number is modified, and every time a device receives a challenge, it checks the

skip number to see if the challenge is valid. For convenience, the trusted computing device may set the skip number to an integer value greater than zero, and the skip number may be decremented at each retransmission. Any device receiving a challenge with a skip number of zero ignores the challenge and does not retransmit the challenge. This prevents the challenge from propagating too widely.

According to a fourth aspect of the present invention, there is provided a method of ascertaining the status of the data processing environment, the method comprising the steps of using a trusted computing device to challenge other devices within a data processing environment, keeping a record of the responses made to the challenges and making the record available.

Preferably the trusted computing device will itself respond to a challenge such that the integrity of the trusted computing device can be verified by the device which challenged it.

According to a further aspect of the present invention, there is provided a method of conducting a transaction in a data processing environment comprising a user device and at least a trusted computing device each having respective communications capabilities, wherein the trusted computing device keeps a record of computing devices that it has identified within the data processing environment, and wherein the user device is arranged to establish communications with the trusted computing device, to receive therefrom at least a portion of the record of computing devices within the data processing environment, and to analyse the record to establish what facilities the user device may access.

Preferably the user device further analyses the record in accordance with a set of security rules contained therein to determine what level of trust the user device can place on the integrity of the data processing environment.

It is thus possible to provide a trusted record of the status and trustworthiness of devices within a data processing network such that a computing device can be spared the task of challenging each device in the computer network in order to ascertain its trustworthiness, but instead can obtain a record of the challenges from a trusted computing device.

Brief Description of the Drawings

The present invention will further be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 is a schematic diagram illustrating the functional components within a trusted computing device;

Figure 2 schematically illustrates the structure of a first data processing environment including a trusted computing device;

Figure 3 schematically illustrates a second data processing environment including a trusted computing device;

Figure 4 is a flow chart illustrating the steps undertaken by a trusted computing device in order to maintain a record of the local data processing environment;

Figure 5 illustrates the data layout of the challenge issued by the trusted computing device; and

Figure 6 is a flow chart illustrating the response of a device in the data processing environment upon receiving a challenge.

Best Mode for Carrying Out the Invention

Many of the concepts underlying trusted computing have already been published. In particular, a specification concerning the functionality of a trusted computing environment has been published by the "trusted computing platform alliance" on their web site at www.trustedpc.org. A trusted computing device of this general type is described in the applicant's co-pending International Patent Application Publication No. PCT/GB00/00528 entitled "Trusted Computing Platform", filed on 15 February 2000, the contents of which are incorporated by reference herein.

In essence, it is desirable to ensure that a computer is trustworthy. In broad terms, this can be achieved by attaching to or embedding in a computer a trusted physical device whose function in relation to that computer is to obtain measurements of data from the computer which provides an integrity metric of the computer platform. The identity of the computing platform and the integrity metric are compared with expected values that are provided by a

trusted party whose role it is to vouch for the trustworthiness of the platform. If the identities and metrics match, then there is a strong implication that the platform is operating correctly. The trusted physical device is normally provided as a tamper evident component such that attempts to interfere with its operation will result in its performance being modified when it is next used.

A trusted platform is illustrated in Figure 1. The platform 2 includes a central processing unit 4, which is in communication with BIOS memory 6, Random Access Memory 8, Read Only Memory 10, a removable data storage device 12, an internal mass storage device 16, at least one communications device 18, a trusted device 20, a video board and associated display 22, and a user input device such as a keyboard 24, via a data bus 26. In a conventional computing device, at power up or reset the CPU initially reads instructions from the BIOS 6. In the early days of computing the BIOS memory which is non-volatile was hard wired and therefore it was not possible to modify its contents. However, with the development of EEPROM it has become possible to modify the BIOS of a computer system. In a trusted computing environment, the boot-up sequence is modified such that the CPU first looks to the trusted device 20 for instructions after reset or power-up. The trusted device 20, having gained initial control of the CPU, then enables the CPU to execute the BIOS program held in the BIOS memory 6. The trusted device can investigate integrity metrics in the BIOS program, such as check sums for the whole or specific part of the BIOS or data at specific addresses in order to determine that the BIOS has not been interfered with or modified. It can compare these values against values certified as being correct by the trusted party. The BIOS 6 may advantageously be contained within the trusted device, thereby ensuring that the BIOS builds the correct environment for the operating system. The trusted device can also, through its interaction with the BIOS, enforce the operation of various security policies. After the BIOS has been loaded, the CPU can then seek to load its operating system from the storage device 16. Once again, the trusted device 20 can challenge the operating system to extract integrity metrics from it and to compare these with metrics provided by the trusted party. Thus, as the system builds up from power-up or reboot the BIOS is first confirmed as trustworthy, and once this has been established tests are made to see that the operating system is trustworthy, and once this has been established further tests may be made to ensure that applications executing on the trusted computer platform are also trustworthy. The trusted computing platform need not

be a general purpose PC wherein applications are loaded from the mass storage device 16 to the random access memory 8, and indeed the trusted device could be an embedded system. In which case, it is likely that application data may also be held in read-only memory 10. The trusted computing device may have a reader 12 for removable media, such as floppy discs, or for interfacing with the smart cards which may be used to help authenticate the identity of a local user who seeks to operate the trusted computing device 2 via its keyboard 24. An interface to the local user is provided via the keyboard 24 and the video display 22, as is well known. The trusted computing device 2 also has a communications device 18 which may support one or more of direct connection with a local area or wide area network, wireless communications, infrared or ultrasonic communication and/or communications with a telecommunication network via a data link.

As shown in Figure 2, the trusted computing device 2 can be a component within a relatively well defined network. Thus other devices such as a printer 30, a scanner 32, and user devices 34 and 36. Each device is connected via a local area network 38. In this environment, the communications medium between devices is well defined and a number of devices on the network can be expected to change only relatively slowly.

A user's device 40, for example in the form of a personal digital assistant can establish communications with the trusted computing device 2 via the communications device 18 of the trusted computing device 2 and also a communications port 42 of the personal digital assistant. The personal digital assistant 40 also includes a central processing unit 44 and a memory 46 holding a set of security rules which the processor 44 uses to decide whether or not it should regard the data processing environment as trustworthy or not.

In use, the trusted computing device 2 challenges the devices 30 to 36 in its computing environment 38 and keeps a record of their responses, including integrity metrics, which enables it to work out whether the devices 30 to 36 can be trusted, that is that they are behaving in an expected manner and have not been subverted by external or malicious processes. The trusted device 2 keeps a record of the results of the challenges in its memory 8 and on the mass storage device 16 (see Figure 1).

When the user's device wishes to use the facilities available of the computing network 39, it establishes communication with the trusted computing device 2, and challenges it in

order to assure itself that the trusted computing device is a trusted computing device and not some rogue device masquerading or spoofing as a trusted computing device. Once the user device 40 has completed the challenge and compared the integrity metric retrieved from the trusted computing device with an expected integrity metric as certified with a trusted authority, the user device 40 may then query the trusted computing device 2 in order to obtain a list of the devices available in the local computing area, the functions that they can perform and whether or not they are trustworthy. Having received this data, the user device 40 then applies the security rules held in the memory 46 in order to determine whether, in accordance with those rules which themselves are defined either by the device owner, administrator or some other person given the rights to modify those rules, whether the local computing environment is trustworthy. If so, the user is informed. Alternatively, the user may be informed if the computing environment is not trustworthy. The user may also be informed which classes of transactions should or should not be undertaken in this environment.

Not all computing environments are as well defined as that shown in Figure 2. Companies may wish to offer computing facilities in publicly accessible environments where it can be expected that most users will establish local wireless communications with some form of gateway or server whilst they are in the environment. Thus, as shown in Figure 3 a trusted computing device 2 may be situated in an environment with a display device 60. Users 62 and 64 having portable computing devices such as personal digital assistants or palm top computers may enter the computing area around the devices 2 and 60 and may establish wireless communications with the trusted computing device 2 in order to enquire what facilities are available in the computing area. The computing device 2 may then inform the devices 62 and 64 about the existence of the display device 60 such that these devices can then interface with it for example to display data which they are not capable of displaying on their own inbuilt display devices. The trusted computing device 2 may also seek to inform each mobile device 62 and 64 about the presence of the other. The trusted computing device may also provide details about access to a telecommunications network 66.

As indicated above, the arrangements of Figures 2 and 3 are simply examples of (relatively simple) data processing environments. Far more complex arrangements involving multiple

network connections may nonetheless be considered data processing environments. In most practical cases, it will be necessary to define a boundary or a way to determine the extent of the environment, but this will generally not be problematic to the person skilled in the art (for example, such a boundary may be formed by a firewall).Figure 4 schematically illustrates a sequence by which the trusted computing device 2 can maintain its record of devices in the data processing environment. Starting at step 70, the trusted computing device 2 listens to data traffic in the computing environment to identify the presence of any new devices. Once the device has been identified, control is passed to step 72 where the trusted computing devices issues a challenge to the new device. Any response made by that device is recorded at step 74, together with the time at which the response was received and then control is passed to step 76 where an analysis of any integrity metric returned by the challenged device is made in order to ascertain whether the device is trustworthy. The result of the analysis is recorded at step 78. The challenged device, or indeed any other device on the network may issue a request to the trusted device to seek information from the record, a test for any such request is made at step 80. If a request has been received, that data is transmitted at step 82, otherwise control is returned to step 70 where the integrity check can be repeated.

The status of the computing environment can be held by the trusted computing device, and consequently displayed by the user devices, in any manner convenient and appropriate for retaining such data. The status information may be provided in the form of a list of computing entities within the computing environment, together with a trust status for each of the computing entities. Most simply, these statuses could be "trusted" (indicating that the computing entity concerned is verified to have the status of a trusted computing device) or "untrusted" (where no such verification has occurred). The "untrusted" status could be broken down further into "not trusted" (where verification has failed) and "untested" (where verification has either not yet occurred or has not taken place sufficiently recently). A further possibility is for there to be multiple levels of trust available for a computing entity, such that the computing entity is able or permitted to perform different functions at the different levels of trust – more and different integrity metrics may be required to determine that a computing entity is at a higher trust level. Such an arrangement is further described in the applicant's copending International Patent Application Publication No. WO 01/27722, the contents of which are incorporated by reference to the fullest extent

permitted by law. In this case, the trusted device 2 should be adapted to determine when these different trust levels are attained, and to indicate these trust levels appropriately when indicating the status of the computing environment.

In a modification of the flow chart showing in Figure 4, steps 70 and 72 may be replaced by a single broadcast challenge. Such a broadcast challenge is schematically shown in Figure 5. The broadcast challenge comprises two parts, the first being the challenge message 90 and the second part being a generation identifier 92.

Devices receiving the challenge shown in Figure 5 may execute the procedure shown in Figure 6. The procedure starts at step 100 where the device receives the challenge. Control is then passed to step 102 where the generation identifier is examined. In a preferred embodiment of the invention, the generation identifier is set to a positive integer number which controls the number of retransmissions which the challenge may undergo. Each time the challenge is retransmitted, the generation identifier is decremented by the device that retransmits the challenge. Thus, once the generation identifier reaches zero the challenge has become "old" and is no longer valid. At step 104 a test is made to see if the generation identifier is greater than zero, if not, control is passed to step 106 where the challenge handling routine is terminated. If the generation identifier is greater than zero control is passed to step 108 where, if the device is programmed to participate in these challenges, it responds to the challenge. From step 108 control is passed to step 110 where the challenge identifier is decremented and then to step 112 where the challenge is retransmitted with the decremented generation identifier. Control is then passed to step 114 which represents the end of this routine.

It is thus possible to provide a measure of the integrity and facilities available within a local, and possibly varying data processing network.